重み付きL∞距離Voronoi図の構成と文字配置問題

3L-6

神代伸彦　　今井 浩

九州大学工学部

1．まえがき

平面上にn個の点が与えられたときこれらの点の勢力圏をあらわすVoronoi図は、計算幾何学の数多くの基本的な問題を解くのに用いられる有益な手法である[3]。各点の勢力圏を定義するときの距離は一般にはEuclid距離を用いるが、その他にもL_1距離やL_∞距離を用いた方が効率よく解ける問題もある。

本稿では、平面上の各点にそれぞれ異なる正の値の重みを持たせてL_∞距離で定義したVoronoi図を構成するアルゴリズムを概説する。これはDavenport-Schinzel列を用いると$O(n^2 \log n)$時間で構成できる。さらに、この図が文字配置問題[5]において地図が動的に変化する場合の簡単なバージョンに適用できることを述べる。

重み付きVoronoi図については、Euclid距離を用いて構成すると、最悪の場合に$O(n^2)$のVoronoi辺、Voronoi点が存在することが示されており、その計算複雑度が$O(n^2)$であることが分かっている[2]。またDavenport-Schinzel列は、関数集合の最小値すなわち下側エンベロープを求めるときに用いられる手法で、例えばn個の関数が2変数関数でそれらが部分的に線形である場合、$O(n^2\alpha(n)\log n)$の手間で下側エンベロープを求めるアルゴリズムが文献[1]で紹介されている。本稿では、これらの特別な場合としてL_∞距離によるVoronoi図および1変数関数のDavenport-Schinzel列を用いてアルゴリズムの計算複雑度の改良を図る。

2．重み付きL_∞距離Voronoi図

点$P_1(x_1, y_2)$と$P_2(x_2, y_2)$のL_∞距離$d_\infty(P_1, P_2)$とは次式で定義された距離である[3]。

$$d_\infty(P_1, P_2) = \max\{|x_1 - x_2|, |y_1 - y_2|\}$$

点P_iに重みw_iを持たせたときの平面上の任意の点Pへの重み付きL_∞距離$d_w(P, P_i)$を次のように定義する。重みの付け方は$w_i \leq w_j (i<j)$とする。

$$d_w(P, P_i) = d_\infty(P, P_i) / w_i$$

これにより点P_iの勢力圏すなわちVoronoi領域$V_w(P_i)$は、

$$V_w(P_i) = \bigcap_{j=1}^{n} \{P \mid d_w(P, P_i) < d_w(P, P_j)\}$$

と表現できる。n個の点について領域$V_w(P_i)$は互いに非連結であることがわかるので、平面はこのVoronoi領域によって分割される。こうして得られた重み付きL_∞距離Voronoi図を以後略してWVDと記述する。

Voronoi図の特徴の1つは、n個の点のうち任意の1点を選んでその点を中心とする極大空円がVoronoi領域に内接することである[3]。同様の特徴がL_∞距離Voronoi図にもあり、この場合極大空円ではなく極大空正方形となる。したがって、n個の各点からそれを中心とする正方形を描き、それを徐々に等縮尺で拡大していったときに別の正方形よりも先に覆うことができる領域がその点のVoronoi領域となる。ところが各点に重みを付けたWVDでは、各正方形の広がり方が一様でなくなり、重みの値の比に比例する速さで拡大していく。そのため一番重みの値が大きい点のVoronoi領域は他のVoronoi領域をすべて包含する無限に広がる平面となり、残りの点のVoronoi領域はすべて閉じた領域となる。

直感的理解を助けるため、WVDを簡単なバージョンで説明する。

n個の点からなる点集合から任意の点対を持ってきて、この2つの点によるWVDを考える。平面上に2点P_1、P_2があり、それぞれの重みをw_1、$w_2 (w_1 < w_2)$とする。この2点によるWVDは図1のようになる。点P_2の正方形S_2の方が点P_1の正方形S_1よりも拡大スピードが速いため、ある時間を過ぎるとS_2がS_1を包含してしまってそれ以降はS_1がいくら大きくなっても自分自身の領域を形成することができなくなる。したがって、P_1のVoronoi領域$V_w(P_1)$は閉じた領域となり、$V_w(P_2)$が残りの平面を覆うことになる。

以上の議論はn個の点すべてを考慮した場合についても適用できる。

3．WVDの構成

WVDすなわち重み付きL_∞距離Voronoi図を構成するときは、前節の議論から正方形の拡大を考えるとよいことが分かる。n個の点が分布しているxy平面とこれに垂直なz軸からなる3次元空間を考える。重みがすべて等しい場合のL_∞距離Voronoi図は、母線がxy平面となす角度が45°の正四角錘（底面が正方形である四角錘）をその頂点がn個の点に一致するように立て、それを$z=-\infty$から眺めた状態を2次元平面に投影した図である。Voronoi辺はこの正四角錘の側面にあたる平面（底辺がない半無限三角形）の交線である。WVDでは各点に重みが付いているので、母線の角度が45°にはならずに重みによって異なる。つまり、重みの値が大きい場合は正四角錘の母線の角度は小さくなり、値が小さい場合は角度は大きくなる。

ここでは、n個の点の中の1点P_iに着目して$V_w(P_i)$を求める方法を述べる。

(1) Voronoi領域の分割

$V_w(P_i)$を2本の45°の直線で4つの領域に分割する。この4つの領域の中の1つに着目して得られる諸性質は、他の3つの領域に関しても共通して得られるということができる。点P_iのVoronoi領域$V_w(P_i)$は点P_iより重みが大きい他の$(n-i)$個の点によって拘束される(図2)。4つの領域の中の1つ（点P_iを頂点とする半無限三角形平面）に着目すると、この領域は最大$(n-i)$本の連続した線で分割される。この領域を点P_iが下にくるような向き

Constructing the Weighted L_∞-Distance Voronoi Diagram and its Application to the Automatic Character Placing Problem

Nobuhiko KOJIRO and Hiroshi IMAI

Faculty of Engineering, Kyushu University

で観察すると、$V_w(P_i)$は$(n-i)$個の連続線の下側の領域の共通部分になる。また、この連続線の下側エンベロープはVoronoi辺に相当する。したがって、$(n-i)$個の連続線の下側エンベロープを求めることで$V_w(P_i)$を求めることが可能となる。これはDavenport-Schinzel列と呼ばれる最小値関数表現法が利用できる。

（２）Davenport-Schinzel列[4]

n個の１変数連続関数$f_i(x)$ $(i=1,\cdots,n)$の集合で、任意の対の関数が互いにたかだかｓ個の点で交わるものを考える。これらの関数の最小値を値としてとる関数をｆとすると、

$$f(x)=\min_{i=1,\cdots,n}\{f_i(x)\}$$

つまり関数ｆのグラフは$f_i(x)$の下側エンベロープである。Davenport-Schinzel列はこの下側エンベロープを、ｘの区間ごとに対応する関数で覚えておく(図３)。

（３）Voronoi辺の複雑度

$V_w(P_i)$の半無限三角形平面を分割する最大$(n-i)$本の連続線の各々は部分的に線形な関数である。しかもそれらは２つの線分からなり、その一方はｘ軸に平行な線分である。したがって、これら$(n-i)$本の線形関数から任意の２つを選んだとき、それらの交点の数は最大２個である。ゆえに$s=2$であるときのDavenport-Schinzel列を利用すればよい。これによると、下側エンベロープを構成する関数の数は$O(n)$であることがわかっている。残りの３つの半無限三角形平面についても同様の議論が成り立つことは明らかである。ゆえにn個の点からなるＷＶＤを構成するVoronoi辺は$O(n^2)$である。

４．ＷＶＤ構成アルゴリズム

本節ではＷＶＤを逐次的に構成するアルゴリズムの概要を述べる。逐次的構成とは、n個の点を重みの値の大きい順に平面に添加していきそのVoronoi領域を順に求めていく構成法である。

step 1.　最も重みの値が大きい点、すなわちP_nを平面に添加する。$V_w(P_n)$は平面全域である。

step 2.　P_{n-1}を平面に添加し、P_nとの境界となる$V_w(P_{n-1})$を構成する。

step 3.　P_iを平面に添加し、$P_n,\cdots,P_{i+1}$との境界となる$V_w(P_i)$を構成する。

(step 1)および(step 2)の計算は定数の手間で可能である。(step 3)においては、１つの点に対するVoronoi領域を構成する線分の数は$O(n)$であることから、各点のVoronoi領域は$O(n\log n)$で計算できる。ゆえに、ＷＶＤを構成する逐次アルゴリズムの計算複雑度は$O(n^2\log n)$である。

５．文字配置問題への応用

ＷＶＤの応用例として、拡大縮小を考慮した文字配置問題がある。文字配置問題[5]とは、地図上のn個の点（施設）に対して文字列を互いに重ならないように配置できるかどうかを判定する問題である。文献[5]は、拡大や縮小を考慮しない場合に$O((n+k)\log n)$の手間で配置判定を行なうアルゴリズムを示している。

地図中に配置する文字列は長方形で代表できるが、この各長方形が正方形になるような変換をする。地図の拡大縮小をこの正方形の見地から眺めると、地図の縮小（拡大）＝正方形の拡大（縮小）であることがいえる。したがって前処理としてＷＶＤを求めておけば、地図を縮小したときに重なりを生じる正方形の組合せをＷＶＤを元にして効率よく求めることができる。

６．まとめ

本稿では、重み付きのL_∞距離Voronoi図（ＷＶＤ）を構成する$O(n^2\log n)$のアルゴリズムについて議論した。またこのVoronoi図を前処理として構成しておくことで、拡大縮小を考慮した文字配置問題の簡単なバージョンへの応用が可能であることを述べた。尚、本研究の一部は、文部省科学研究費の援助を受けている。

参考文献

[1]H. Edelsbrunner, J. Pach, J. T. Schwarz and M.Sharir : “On the Lower Envelope of Bivariate Functions and its Applications”, Proc. 28th IEEE Annual Symp. on Foundations of Computer Science, pp.27-37, 1987.

[2]F.Aurenhammer and H.Edelsbrunner : “An Optimal Algorithm for Constructing the Weighted Voronoi Diagram in the Plane”, Pattern Recognition, Vol.17, No.2, pp.251-257, 1984.

[3]伊理正夫監修, 腰塚武志編集 : “計算幾何学と地理情報処理”, bit別冊, 共立出版, 東京, 1986.

[4]S.Hart and M.Sharir : “Nonlinearity of Davenport-Schinzel Sequences and of Generalized Path Compression Schemes”, Combinatorica, Vol.6, pp.151-177, 1986.

[5]今井 浩, 神代伸彦, 青沼裕美 : “点図形に対する文字配置アルゴリズムとその応用例”, 情報処理学会九州支部研究会報告, pp.11-20, 1989.

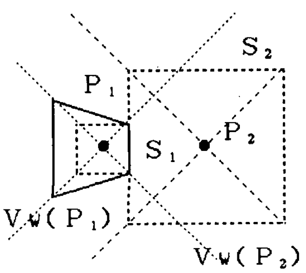

図１．２点によるＷＶＤ

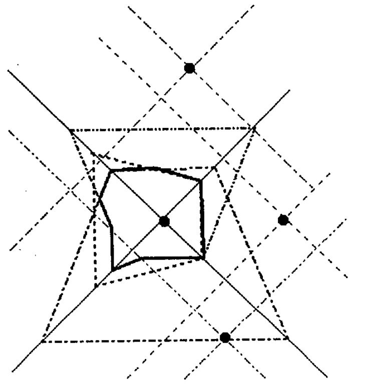

図２．他の３点との境界の共通部分

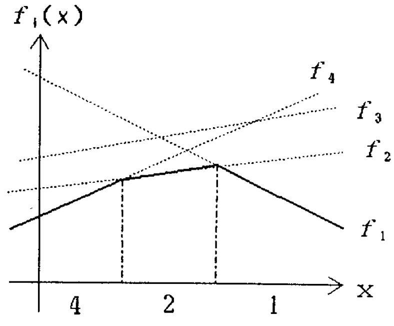

図３．線形関数のDavenport-Schinzel列